


Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký, Chức vụ và Họ tên)	(Ngày duyệt đề)
--	--------------	--	-----------------

(phần phía trên cần che đi khi in sao đề thi)

 TRƯỜNG ĐH BÁCH KHOA – ĐHQG-HCM KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH	THI CUỐI KỲ		Học kỳ/năm học		2	2021-2022
			Ngày thi		17/05/2022	
	Môn học	Lập Trình Nâng Cao				
	Mã môn học	CO2039				
	Thời lượng	90 phút	Mã đề			
Ghi - Không sử dụng tài liệu chú: - Nộp lại đề thi cùng với bài làm						

GỢI Ý TRẢ LỜI

Câu 1 (L.O.1): Sinh viên sử dụng các kiến thức được cung cấp trong môn học để phân biệt các cặp thuật ngữ sau:

- (i) Lớp (*Class*) và Đối tượng (*Object*)
- (ii) Trừu tượng hoá dữ liệu (*data abstraction*) và Bao đóng dữ liệu (*data encapsulation*)
- (iii) Thừa kế (*inheritance*) và đa hình (*polymorphism*)

Câu 2 (L.O.1): Các tầm vực (*Access Modifier*) có thể cài đặt cho các thuộc tính và tác vụ trong các class: Public, Protected, Private. Sinh viên giải thích ý nghĩa của các tầm vực này. Sinh viên nên viết 1 chương trình ví dụ.

Câu 3 (L.O.2, L.O.4): Sinh viên sử dụng các kiến thức về OOP đã học để viết chương trình C++

Hãy tham khảo các hiện thị của 1 chương trình ví dụ như sau (trong đó các thông tin sau dấu ? là do người sử dụng nhập vào):

```

Nhap so sinh vien? 1
-----
Nhap thong tin Sinh vien 1:
Student ID? 1234
Name? NguyenVanA
-----
Sinh vien 1 co bao nhieu Grades? 2
Name cua Grade 1? OOP
Mark cua Grade 1: 9
-----
Name cua Grade 2: FP
Mark cua Grade 2: 10
=====
Chon Sinh vien de tinh diem trung binh? 1
Diem trung binh cho Sinh vien 1: 9.5
  
```

Câu 4 (L.O.1.3): Sự khác nhau giữa Lập trình theo hướng đối tượng (*Object-Oriented Programming - OOP*) và theo hướng hàm (*Functional Programming - FP*): sinh viên có thể đề cập tới các thành phần cơ bản của OOP là object/class, của FP là hàm. Sinh viên nên cho 1 ví dụ

Câu 5 (L.O.3):

- (i) Biểu thức Lambda (*Lambda expression*) được dùng để định nghĩa 1 hàm mà không cần khai báo tên (hàm dùng 1 lần). Sinh viên nên cho 1 ví dụ
- (ii) Hàm Higher-Order (*function*) là hàm nhận 1 hàm nào đó làm tham số. Sinh viên nên cho 1 ví dụ

Câu 6 (L.O.3):

(i) Hàm `ztn` tính số lần xuất hiện của số nguyên trong 1 danh sách

(ii) Kết quả của các lệnh chạy sau

`ztn [1,2,3] 2` Lỗi

`ztn [1,2,3,2] 2` Lỗi

`ztn ([1,2,3],2)` 2

`ztn ([1,2,3,2],2)` 2

Câu 7 (L.O.4): Sử dụng kiến thức về viết script được cung cấp trong khóa học để viết hàm (Lưu ý có nhiều cách định nghĩa hàm)

Ví dụ: `sum_square [1..10]` sẽ trả về kết quả $1 + 4 + 9 == 14$

--- HẾT ---

